

Model Multi Layer Perceptron untuk Indoor Positioning System Berbasis Wi-Fi

Yuan Lukito *)

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana
Jl. Dr. Wahidin Sudiro Husodo 5-25, Yogyakarta 55224, Indonesia

Abstract – *Indoor positioning system issue is an open problem that still needs some improvements. This research explores the utilization of multi layer perceptron in determining someone's position inside a building or a room, which generally known as Indoor Positioning System. The research was conducted in some steps: dataset normalization, multi layer perceptron implementation, training process of multi layer perceptron, evaluation and analysis. The training process has been conducted many times to find the best parameters that produces the best accuracy rate. The experiment produces 79,16% as the highest accuracy rate. Compared to previous research, this result is comparably lower and needs some parameters tweaking or changing the neural networks architectures.*

Keywords – *Indoor positioning; Wi-Fi; Multi-layer perceptron*

Abstrak – *Masalah penentuan posisi di dalam ruangan masih memerlukan banyak perbaikan. Penelitian ini mencoba melakukan eksplorasi terhadap penggunaan multi layer perceptron untuk penentuan posisi seseorang di dalam gedung atau ruangan, yang lebih dikenal dengan istilah Indoor Positioning System. Penelitian ini dilaksanakan dalam beberapa tahap yaitu normalisasi dataset, implementasi multi layer perceptron, pelatihan multi layer perceptron dan proses pengujian serta analisis. Proses pelatihan dilakukan beberapa kali untuk menemukan parameter-parameter yang menghasilkan akurasi terbaik. Dari hasil percobaan yang dilakukan didapatkan tingkat akurasi terbaik sebesar 79,16%. Hasil tersebut masih lebih rendah jika dibandingkan dengan hasil penelitian sebelumnya, sehingga memerlukan perubahan pengaturan parameter atau pengubahan arsitektur jaringan syaraf tiruan yang digunakan.*

Kata Kunci – *Posisi dalam ruangan; Wi-Fi; Multi layer perceptron*

I. PENDAHULUAN

Indoor Positioning System (IPS) dapat dianalogikan seperti *Global Positioning System* (GPS), namun diimplementasikan untuk keperluan penentuan

posisi dan petunjuk navigasi di dalam suatu ruangan tertutup atau suatu gedung. Sinyal GPS umumnya sulit diterima di dalam sebuah gedung, terutama pada gedung bertingkat banyak. IPS dikembangkan untuk mengatasi permasalahan tersebut. Jika GPS memanfaatkan satelit, maka IPS diimplementasikan menggunakan perangkat tambahan yang dipasang khusus, misalnya *bluetooth* [1], *Radio-frequency Identification* (RFID) [2], [3], *Light Emitting Diode* (LED) [4], pendekatan berbasis video [5], pemanfaatan sinyal *Wi-Fi* [6], [7], RFID [8] dan gelombang ultrasonik [9].

Saat ini, banyak gedung perkantoran yang menyediakan layanan akses Internet dengan memasang *Wi-Fi access point* (WAP) di beberapa lokasi di dalam gedung. Umumnya penempatan WAP tersebut dilakukan dengan pertimbangan tertentu agar cakupannya dapat meliputi sebagian besar gedung, sehingga setiap orang dapat mengakses Internet, dimanapun posisinya di dalam gedung. Selain untuk mengakses Internet, fasilitas WAP tersebut juga dapat dimanfaatkan untuk implementasi IPS.

Informasi yang dapat diambil dari sebuah WAP adalah *Service Set Identifier* (SSID), *channel* yang digunakan untuk mengirimkan sinyal Wi-Fi, sifat otentikasi (*open access*, *secure WPA* maupun *WPS*), *mac address* dari *access point* dan kekuatan sinyal yang diterima (dalam satuan dBm). Kekuatan sinyal yang diterima akan berubah-ubah sesuai dengan jarak penerima sinyal terhadap WAP dan kondisi topologi ruangan (misalnya ada tembok atau penghalang). Tingkat kekuatan sinyal yang diterima (*received signal strength*, RSS) dapat dimanfaatkan untuk menentukan posisi seseorang.

Metode yang umumnya digunakan untuk menentukan posisi dengan menggunakan informasi RSS adalah *triangulation*. Metode ini memanfaatkan perhitungan jarak dari penerima sinyal Wi-Fi terhadap minimum tiga WAP terdekat. Perhitungan jarak dapat menggunakan teknik *time on arrival* (TOA), *time difference of arrival* (TDOA) atau *roundtrip time of flight* (RTOF).

Metode penentuan posisi berdasarkan jarak tersebut memiliki beberapa kelemahan, yaitu antara lain perlunya perencanaan penempatan WAP, kekuatan sinyal yang berbeda jika merek atau jenis WAP yang digunakan berbeda, serta sangat ditentukan oleh bentuk gedung. Penempatan WAP harus dilakukan untuk menunjang teknik *triangulation*, padahal saat ini sudah banyak yang

*) Penulis korespondensi (Yuan Lukito)
Email: yuanlukito@ti.ukdw.ac.id

memasang WAP hanya untuk keperluan akses Internet, sehingga perlu perubahan peletakan WAP yang tentunya membutuhkan tenaga, biaya dan waktu yang cukup banyak. Masalah berikutnya adalah kekuatan sinyal yang berbeda-beda untuk beberapa WAP yang merek dan jenisnya berbeda. Saat ini terdapat beberapa standar seperti 802.11a, 802.11b, 802.11g, 802.11n dan 802.11ac yang berbeda-beda jangkauan dan kekuatan sinyalnya. Untuk mengatasi masalah ini diperlukan langkah-langkah untuk melakukan kalibrasi dan normalisasi supaya setiap WAP memiliki kondisi yang setara. Masalah terakhir adalah bentuk gedung dan ruangan yang berbeda-beda akan sangat menyulitkan dalam menentukan posisi ideal WAP untuk teknik *triangulation*.

Penentuan posisi sebenarnya dapat dianalogikan seperti proses klasifikasi, dimana posisi seseorang dapat ditentukan berdasarkan informasi RSS yang didapatkan. Pendekatan ini telah diterapkan pada [10] dengan menggunakan metode *K-Nearest Neighbors (KNN)*, *Naïve Bayes*, *J48* dan *Support Vector Machine (SVM)*. Dari penelitian tersebut didapatkan metode KNN menghasilkan tingkat akurasi terbaik, yaitu sebesar 83,58%. Pendekatan klasifikasi posisi seperti ini memiliki kelemahan yaitu tidak bisa didapatkan informasi posisi dalam bentuk koordinat, baik koordinat dua dimensi atau tiga dimensi. Posisi yang didapatkan dari pendekatan klasifikasi ini adalah informasi ruangan terdekat, misalnya laboratorium perangkat bergerak, ruang rektorat dan ruang rapat.

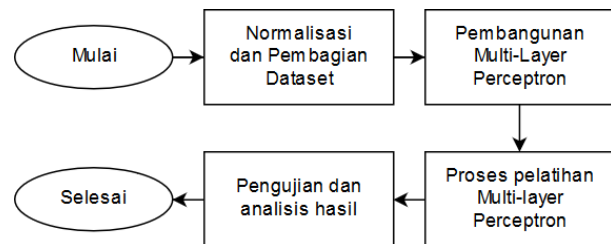
Dalam penelitian ini dilakukan eksplorasi penggunaan jaringan syaraf tiruan untuk keperluan menentukan posisi berdasarkan kekuatan sinyal Wi-Fi yang diterima dari beberapa WAP yang terdekat. Metode jaringan syaraf tiruan dipilih dengan beberapa pertimbangan, antara lain: (1) dapat diterapkan dalam berbagai macam permasalahan, terutama dalam masalah klasifikasi, (2) relatif mudah digunakan dan diimplementasikan, terutama dengan banyaknya pustaka-pustaka pendukung dalam berbagai macam bahasa pemrograman, (3) dapat menghasilkan tingkat akurasi yang lebih baik daripada metode-metode lainnya, jika dilihat dari kesuksesan *deep learning* (yang basisnya adalah jaringan syaraf tiruan) dalam memecahkan masalah-masalah klasifikasi. Penelitian ini mengembangkan model *multi layer perceptron* (MLP) untuk klasifikasi posisi. Dataset yang digunakan sama seperti yang digunakan pada penelitian [10], sehingga hasilnya dapat langsung dibandingkan dengan metode-metode lainnya yang telah diujikan sebelumnya.

II. METODOLOGI PENELITIAN

Dataset RSS Wi-Fi yang digunakan dalam penelitian ini memiliki format seperti pada Tabel 1. Dataset RSS Wi-Fi yang digunakan memiliki 178 atribut, dengan atribut pertama adalah nama ruangan, dan atribut-atribut berikutnya adalah nilai kekuatan sinyal Wi-Fi dari WAP tertentu dalam satuan dBm. Dataset tersebut berisi

Tabel 1. Format dataset RRS Wi-Fi

Ruangan	AP-1	AP-2	AP-3	...	AP-177
Rektorat	-89	-100	-92	...	-84
Perpustakaan	-78	-80	-100	...	-100
...
Atrium D	-82	-100	-100	...	-89



Gambar 1. Tahapan penelitian

11654 data yang dibagi menjadi 42 ruangan. Atribut ruangan akan digunakan sebagai *class*, sehingga hasil klasifikasi berada dalam cakupan 42 *class* yang ada. Seluruh *class* tersebut merupakan ruangan-ruangan yang dapat diakses oleh orang umum, yang tersebar dari 9 gedung yang ada di lingkungan kampus Universitas Kristen Duta Wacana (UKDW).

Secara umum penelitian ini merupakan penelitian pengembangan model klasifikasi menggunakan jaringan syaraf tiruan berbasis eksperimen yang dilakukan dalam langkah-langkah seperti pada Gambar 1. Pada tahap normalisasi dan pembagian dataset, dilakukan normalisasi nilai-nilai pada dataset menjadi nilai-nilai di dalam rentang 0 sampai 1. Tahap pembangunan model multi layer perceptron dilakukan dalam bahasa pemrograman Python dengan bantuan pustaka Tensorflow [11]. Pada tahap pelatihan model, dilakukan proses pelatihan beberapa kali sesuai dengan parameter-parameter yang diujikan. Pengujian dan analisis dilakukan terhadap model yang didapatkan dari hasil pelatihan. Hasil pengujian akan dibandingkan dengan hasil dari penelitian sebelumnya.

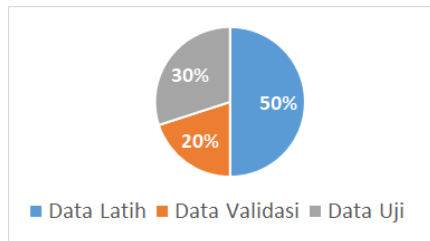
Penelitian diawali dengan proses normalisasi data, yang dilakukan dengan cara melakukan konversi nilai kekuatan sinyal yang berada dalam rentang -100 dBm (nilai sinyal terlemah) sampai -40 dBm (nilai sinyal terkuat) menjadi nilai dalam rentang 0 sampai 1. Normalisasi dilakukan dengan Persamaan (1). Hasil normalisasi dari dataset yang dilakukan pada data Tabel 1 dapat dilihat pada Tabel 2.

$$RSSn_i = \frac{(RSS_i + 100)}{60} \quad (1)$$

Setelah proses normalisasi, berikutnya adalah proses pemisahan dataset menjadi tiga bagian secara acak, yaitu data latih, data validasi dan data uji. Pada penelitian ini digunakan proporsi data latih:validasi:uji sebesar 50% : 20% : 30%, seperti ditunjukkan pada Gambar 2.

Tabel 2. Contoh hasil normalisasi dari data Tabel 1

Ruangan	AP-1	AP-2	AP-3	...	AP-177
Rektorat	0.183	0	0.133	...	0.267
Perpustakaan	0.367	0.333	0	...	0
...
Atrium D	0.300	0	0	...	0.183



Gambar 2. Proporsi data latih, validasi dan uji

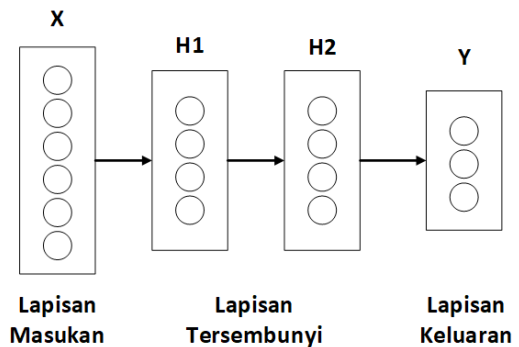
Tabel 3. Parameter MLP yang digunakan

Parameter	Nilai
Jumlah lapisan	4 (1 masukan, 2 tersembunyi dan 1 keluaran)
Jumlah neuron pada lapisan masukan	177 (sesuai dengan jumlah atribut RSS pada dataset)
Jumlah neuron pada lapisan tersembunyi	Dalam rentang 64 – 384.
Jumlah neuron pada lapisan keluaran	42 (sesuai dengan jumlah class pada dataset)
Learning rate	0.001 (menggunakan AdamOptimizer)
Jumlah epoch maksimum	10000 epochs
Nilai cost/error minimum	0,010, 0,025, 0,050

Data latih digunakan untuk proses pelatihan MLP, sedangkan data validasi digunakan untuk melakukan pengecekan proses pelatihan. Data uji digunakan untuk mengukur tingkat akurasi dari MLP yang dihasilkan dari proses pelatihan. Implementasi MLP menggunakan pustaka Tensorflow. Parameter-parameter yang digunakan dalam MLP yang dibangun dapat dilihat pada Tabel 3. Secara umum skema arsitektur MLP yang diimplementasikan terdiri dari 4 lapisan yang saling terhubung sepenuhnya (*fully connected layers*), seperti yang dapat dilihat pada Gambar 3.

Pembentukan MLP diawali dengan inisialisasi nilai *weight* dan bias dari hubungan setiap neuron dengan nilai acak (dalam rentang -1 sampai 1). Langkah berikutnya adalah menghubungkan lapisan masukan dengan lapisan tersembunyi dan terakhir menghubungkan lapisan tersembunyi ke lapisan keluaran. Nilai yang keluar dari lapisan tersembunyi didapatkan dari Persamaan (2).

$$H_1 = \text{ReLU} \left(\sum_{i=1}^n w_i \times X_i + b_i \right) \quad (2)$$



Gambar 3. Arsitektur MLP

Tabel 4. Parameter dalam percobaan

Cost/Error minimum	Jumlah neuron pada lapisan tersembunyi
0,010,	32, 64, 96, 128, 160, 192,
0,025,	224, 256, 288, 320, 352,
0,050	384

Persamaan (2) menggunakan fungsi *Rectified Linear Unit (ReLU)* seperti pada Persamaan (3). Penggunaan fungsi ReLU untuk menghindari nilai negatif. Untuk lapisan tersembunyi yang kedua (H2) juga didapatkan dengan menggunakan Persamaan (2).

$$\text{ReLU} \mid X = \begin{cases} 0 & \text{jika } X < 0 \\ X & \text{jika } X \geq 0 \end{cases} \quad (3)$$

Untuk mendapatkan hasil di lapisan keluaran dilakukan dengan perhitungan aktivasi linier seperti pada Persamaan (4). Data pelatihan, validasi dan uji semuanya merupakan data yang sudah memiliki label (class) yang dibuat dalam bentuk one-hot-encoding. Hasil klasifikasi ditentukan berdasarkan nilai neuron terbesar pada lapisan keluaran.

$$Y_i = \sum_{i=1}^n w_i \times h_i + b_i \quad (4)$$

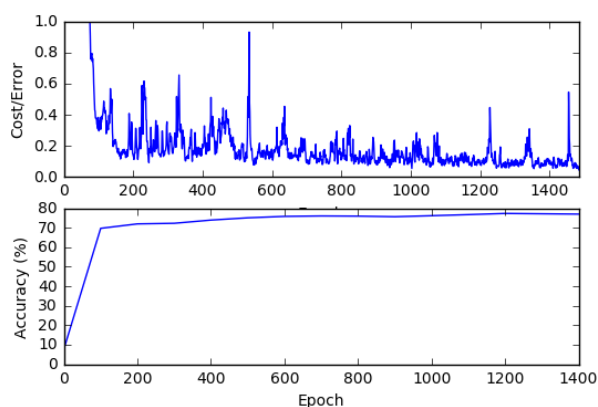
Pembangunan model MLP dilakukan dalam bahasa pemrograman Python dengan bantuan pustaka Tensorflow. Dengan menggunakan pustaka Tensorflow maka proses pembangunan model MLP dapat dilakukan dengan cepat. Dataset dan *script Python* yang digunakan dalam penelitian ini dapat diakses di <https://github.com/yuanlukito-ti/WIFI-MLP>.

Proses pengujian dilakukan dalam beberapa percobaan proses pelatihan MLP dengan beberapa parameter yang berbeda-beda seperti ditunjukkan pada Tabel 4. Pada tahap pengujian ini, evaluasi model yang terbentuk dilakukan berdasarkan akurasi yang didapatkan berdasarkan persamaan (5).

$$\text{Akurasi}(\%) = \frac{\text{Jumlah benar}}{\text{jumlah data uji}} \times 100\% \quad (5)$$

Tabel 5. Hasil pengujian *error* minimum 0,05

Jumlah Neuron	Tingkat akurasi (%)	Epochs
32	72,46	1253
64	73,34	472
96	72,69	357
128	70,58	254
160	74,13	547
192	72,34	363
224	72,51	294
256	72,12	234
288	77,85	1280
320	75,98	685
352	78,48	1484
384	78,16	2319



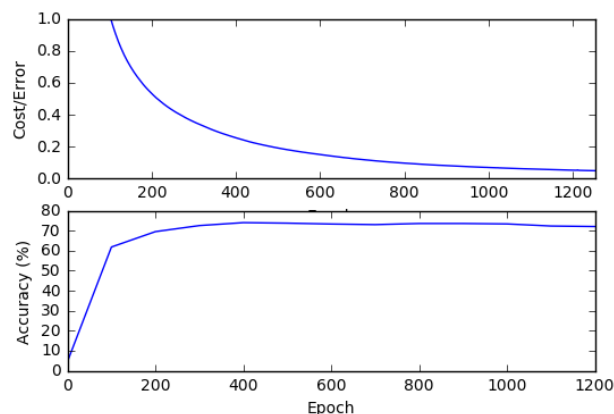
Gambar 4. Laju perkembangan nilai *error* dan tingkat akurasi pada *error* minimum 0,05 dan neuron sebanyak 352

III. HASIL DAN PEMBAHASAN

Percobaan dilakukan dengan mengubah-ubah nilai *error* minimum dan jumlah neuron pada lapisan tersembunyi. Secara keseluruhan terdapat 36 variasi parameter yang diujikan. Hasil pengujian dengan menggunakan parameter *error* minimum 0,05 dapat dilihat pada Tabel 5.

Tingkat akurasi yang dihasilkan pada nilai *error* minimum 0,05 berada dalam rentang 70,58% - 78,48%. Jumlah neuron di lapisan tersembunyi sebanyak 352 memberikan tingkat akurasi tertinggi sebesar 78,48% jika menggunakan nilai *error* minimum 0,05. Laju perkembangan nilai *error* dan tingkat akurasi dari data validasi pada parameter *error* minimum 0,05 dan jumlah neuron sebanyak 352 dapat dilihat pada Gambar 4.

Dari Gambar 4 dapat dilihat perubahan nilai *error* yang sangat fluktuatif, berubah-ubah dengan cepat. Hal ini disebabkan oleh jumlah neuron pada lapisan tersembunyi yang semakin banyak sehingga menambah kompleksitas proses pelatihan, terutama pada bagian perubahan bobot untuk setiap neuronnya. Laju perkembangan nilai *error* dan tingkat akurasi untuk jumlah neuron yang lebih sedikit dapat dilihat pada Gambar 5 sebagai perbandingan.



Gambar 5. Laju perkembangan nilai *error* dan tingkat akurasi pada *error* minimum = 0,05 dan neuron = 32

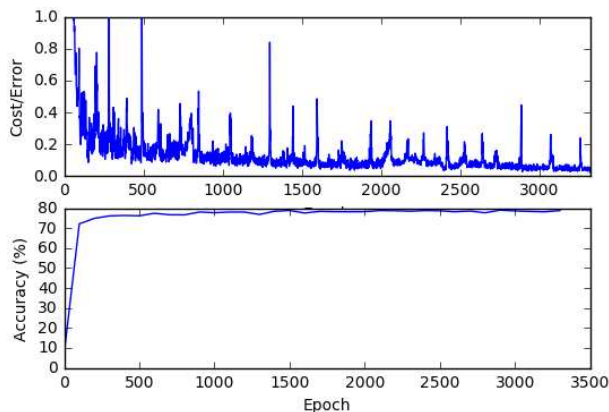
Tabel 6. Hasil pengujian *error* minimum 0,025

Jumlah Neuron	Tingkat akurasi (%)	Epochs
32	69,79	3360
64	73,90	1251
96	74,39	848
128	71,52	886
160	72,89	452
192	74,81	849
224	76,03	1525
256	77,06	2151
288	77,60	2559
320	76,68	3050
352	77,71	2934
384	79,16	3318

Dari Gambar 5 dapat dilihat bahwa jumlah neuron yang lebih sedikit menyebabkan proses pelatihan relatif lebih mudah dan lebih singkat jika dibandingkan dengan menggunakan jumlah neuron yang lebih besar. Perubahan nilai *error* yang didapatkan juga relatif lebih halus dan tidak terjadi perubahan yang fluktuatif. Hasil percobaan dengan nilai *error* minimum 0,025 dapat dilihat pada Tabel 6.

Pada parameter *error* minimum sebesar 0,025 didapatkan tingkat akurasi pada rentang 69,79% - 79,16%. Akurasi tertinggi sebesar 79,16% didapatkan pada penggunaan neuron sebanyak 384 pada lapisan tersembunyi. Laju perkembangan nilai *error* dan tingkat akurasi validasi dapat dilihat pada Gambar 6.

Dari Gambar 6 dapat dilihat bahwa sama seperti hasil pada Gambar 4, jumlah neuron yang semakin besar dapat menambah kompleksitas proses pelatihan sehingga terjadi perubahan yang fluktuatif dari nilai *error*. Namun, secara umum jumlah neuron yang semakin besar dapat meningkatkan tingkat akurasi yang didapatkan. Hasil percobaan dengan nilai *error* minimum sebesar 0,01 dapat dilihat pada Tabel 7.



Gambar 6. Laju perkembangan nilai *error* dan tingkat akurasi validasi pada *error* minimum 0,025 dan neuron=384

Tabel 7. Hasil percobaan *error* minimum 0,01

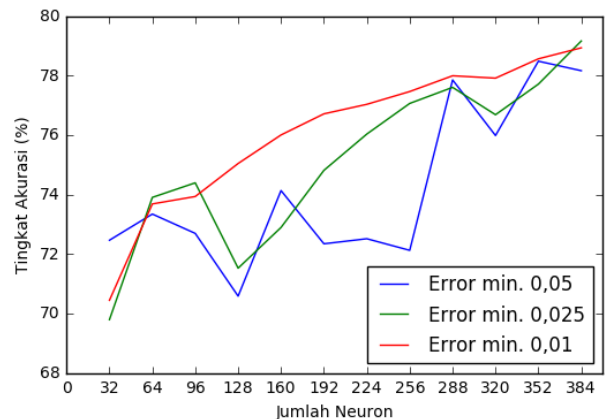
Jumlah Neuron	Tingkat akurasi (%)	Epochs
32	70,44	10000
64	73,68	10000
96	73,93	3246
128	75,04	4835
160	76,00	4650
192	76,71	7508
224	77,03	5193
256	77,46	6069
288	77,99	5292
320	77,91	8040
352	78,56	5425
384	78,93	7440

Pada parameter nilai *error* minimum sebesar 0,01 didapatkan tingkat akurasi pada rentang 70,44% - 78,93%. Akurasi tertinggi didapatkan pada jumlah neuron pada layar tersembunyi sebanyak 384 neuron, yaitu sebesar 78,93%. Dengan tingkat *error* minimum yang semakin kecil, proses pelatihan menjadi relatif lebih lama, sehingga membutuhkan jumlah *epoch* yang lebih banyak. Hasil percobaan secara keseluruhan dapat dirangkum menjadi satu, seperti yang dapat dilihat pada Gambar 7. Model ini sudah cukup baik untuk dapat melakukan IPS dengan menggunakan sinyal Wi-Fi seperti [6] dan [7].

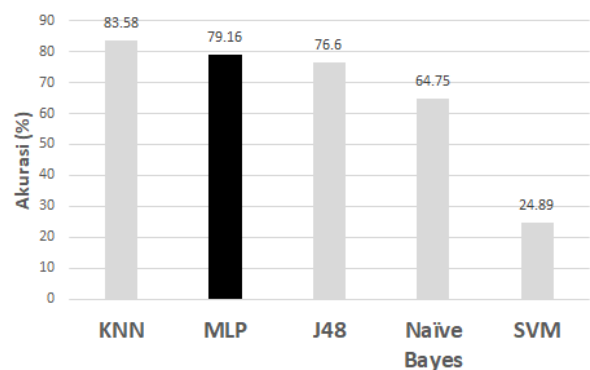
Dari Gambar 7 dapat diambil kesimpulan semakin banyak jumlah neuron pada lapisan tersembunyi, maka tingkat akurasi yang didapatkan cenderung semakin tinggi. Nilai *error* minimum sebesar 0,01 menghasilkan peningkatan akurasi yang lebih stabil dibandingkan dengan nilai *error* minimum lainnya.

Hasil akurasi yang didapatkan dari hasil pengujian tersebut masih lebih rendah jika dibandingkan dengan metode-metode lain yang telah dilakukan dari penelitian sebelumnya [10]. Perbandingan tingkat akurasi yang didapatkan dapat dilihat pada Gambar 8.

Hasil akurasi yang didapatkan dari percobaan dengan menggunakan metode MLP relatif lebih baik dari metode J48, Naïve Bayes maupun SVM. Metode KNN



Gambar 7. Hasil percobaan secara keseluruhan



Gambar 8. Perbandingan hasil percobaan dengan penelitian sebelumnya [10]

mampu menghasilkan akurasi sebesar 83,58% sehingga masih lebih baik dari metode MLP yang digunakan pada penelitian ini. Untuk meningkatkan akurasi dengan metode MLP dapat dilakukan dengan cara melakukan percobaan dengan mengubah nilai parameter-parameter yang digunakan, misalnya dengan menambah jumlah lapisan tersembunyi, menambah jumlah neuron di dalam setiap lapisan maupun juga dapat ditingkatkan dengan menambah jumlah data latih. Alternatif lain yang dapat dilakukan adalah dengan mengubah MLP menjadi *recurrent neural network* (RNN), *convolutional neural networks* (CNN) atau variasi arsitektur jaringan syaraf tiruan lainnya. Penggunaan filter di aplikasi, seperti filter Kalman [1], dapat diterapkan untuk meningkatkan akurasi.

IV. KESIMPULAN

Dari percobaan-percobaan yang telah dilakukan dapat disimpulkan bahwa model *multi layer perceptron* yang dihasilkan dari penelitian sudah cukup baik, dengan tingkat akurasi tertinggi sebesar 79,16% dengan parameter nilai *error* minimum sebesar 0,025 dan jumlah neuron pada layar tersembunyi sebanyak 384 neuron. Hasil akurasi yang didapatkan dari model MLP yang dikembangkan relatif lebih rendah jika dibandingkan

dengan metode KNN yang digunakan pada penelitian sebelumnya. Untuk perbaikan di masa mendatang disarankan untuk melakukan percobaan tambahan dengan mengubah nilai parameter-parameter yang digunakan atau menggunakan arsitektur jaringan syaraf tiruan lainnya.

UCAPAN TERIMA KASIH

Penelitian ini dapat terlaksana berkat dukungan dari Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana, Yogyakarta.

DAFTAR PUSTAKA

- [1] A. Ozer and E. John, "Improving the Accuracy of Bluetooth Low Energy Indoor Positioning System Using Kalman Filtering," in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2016.
- [2] W. An, Z. Shen and J. Wang, "Compact Low-Profile Dual-Band Tag Antenna for Indoor Positioning Systems," *IEEE Antennas and Wireless Propagation Letters*, vol. 16, pp. 400-403, 2017.
- [3] U. Miksaj and D. Bonefacic, "Propagation characteristics of UHF radiofrequency identification system signal for application in indoor positioning," in *2016 22nd International Conference on Applied Electromagnetics and Communications (ICECOM)*, 2016.
- [4] Y. See, N. Noor and Calvin Tan Y.M, "Investigation of Indoor Positioning System using Visible Light Communication," in *2016 IEEE Region 10 Conference (TENCON)*, 2016.
- [5] L. Chen, Chi-Ren Chen and Da-En Chen, "VIPS: A video-based indoor positioning system with centimeter-grade accuracy for the IoT," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017.
- [6] A. Ismail, H. Kitagawa, R. Tasaki and K. Terashima, "WiFi RSS fingerprint database construction for mobile robot indoor positioning system," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016.
- [7] C. Ko and S. Wu, "A Proactive Indoor Positioning System in Randomly Deployed Dense WiFi Networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016.
- [8] L. Mainetti, L. Patrono and I. Sergi, "A Survey On Indoor Positioning System," in *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2014.
- [9] X. Chen and Z. Gao, "Indoor Ultrasonic Positioning System of Mobile Robot Based On TDOA Ranging and Improved Trilateral Algorithm," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, 2017.
- [10] Y. Lukito, and A. R. Chrismanto, "Perbandingan Metode-Metode Klasifikasi Untuk Indoor Positioning System," *Jurnal Teknologi Informasi dan Sistem Informasi*, vol. 1, no. 2, pp. 123-131, 2015.
- [11] M. Abadi et al., "TensorFlow: A System for Large Scale Machine," in *12th USENIX Symposium on Operating System Design and Implementation (OSDI 16)*, 2016.